CLAIMS

WHAT IS CLAIMED IS:

1. A method of processing network data in a network processor comprising:
scheduling a first thread to process a first incoming block of data; and
scheduling a second thread to process a second incoming block of data prior to the first thread completing.

2. The method of claim 1, wherein the first incoming block of data and the second incoming block of data are from a common data packet.

3. The method of claim 2 further comprising:
saving state information by the first thread; and
retrieving the state information by the second thread.

4. The method of claim 3, wherein the state information includes a pointer into a memory indicating where to move the first and second incoming blocks of data.

5. The method of claim 4 further comprising:
storing data to memory in a sequential ordering based on the state information.

6. The method of claim 5 further comprising:

providing the state information to transmit circuitry.

7. A method of processing a data packet received over a network comprising:

processing a first portion of the data packet using a first thread; and

simultaneously processing a second portion of the data packet using a second thread.

8. The method of claim 7 wherein the first thread and the second thread do not time share processing with one another.

9. The method of claim 8 wherein the first thread and the second thread operate out of different microengines.

10. The method of claim 7 wherein the first thread and the second thread time share processing with one another.

11. The method of claim 10 wherein the first thread and the second thread operate out of a common microengine.

12. The method of claim 7 further comprising:

simultaneously with processing the first portion and the second portion of the data packet, processing a third portion of the data packet using a third thread.

13. The method of claim 12 wherein the first thread, the second thread, and the third thread run the same code.

14. The method of claim 13 wherein the first thread, the second thread, and the third thread do not time share processing with one another.

15. An article comprising a computer-readable medium which store computer-executable instructions for receiving data from a plurality of ports, the instructions causing a computer to:

process a first portion of a data packet using a first thread; and

process a second portion of the data packet using a second thread, wherein there is no time sharing between the first thread and the second thread.

16. The article of claim 15, the article further comprises instructions to:

save state information of the first thread; and

restore the state information by the second thread.

17. The article of claim 16, the article further comprises instructions to:

provide the state information to transmit circuitry when an end of packet is detected by a subsequent thread.